

JAP20 Rec'd PCT/PTO 10 FEB 2006

## METHOD AND SYSTEM FOR DEVICE-INDEPENDENT DETERMINATION OF COORDINATES OF A POINT DISPLAYED BY MEANS OF A MICROSCOPE

**[0001]** The present invention relates to a method and a system for the non-instrument-dependent determination of the coordinates of a point imaged using a microscope, and a calibration slide for use therefor. Microscopes are frequently used for detecting small structures which cannot be made out with the naked eye and for discovering characteristic features in such structures. A fundamental use for microscopy in cytology, histology and pathology is for reviewing a specimen and looking for structures, cells or combinations of cells and the like which are of interest. If the sites of these structures are found on the specimen, it is desirable to note them, for a variety of reasons. For example, the structure found has to be located again at a later stage by the same or a different user for checking, for further inspection or for quality control. For this purpose, many microscopes have a unit for determining the coordinates of positions of a point in an instrument-dependent system of coordinates. By electromechanically determining these coordinates, it is possible to return to the location discovered at a later stage.

**[0002]** As a rule, however, the coordinates are instrument-dependent, i.e. the coordinates for this instrument can only be accurately reproduced if no changes have been made to the adjustment of the microscope and no tolerances were present. However, if for example the microscope stage is taken away for repair and reinstalled, it will give different coordinates for the same location on the specimen from those originally determined. Moreover, the coordinate systems of different microscopes, even of the same type, are not (exactly) the same.

**[0003]** There is a need to provide interoperability between any two microscopes, so that, for example, a second user can access, on his system, areas on a specimen which a first user has identified and stored.

**[0004]** This problem is solved by a method and system according to the invention for non-instrument-dependent determination of coordinates of a point imaged using a microscope. The method according to the invention envisages that first of all, at given object-related reference coordinates ( $X_1, Y_1, Z_1$ ) of at least one reference point  $E_1$  in a DICOM coordinate system, the relevant instrument coordinates ( $x_1, y_1, z_1$ ) of the minimum of one imaged reference point  $E_1$  in an instrument-dependent coordinate system are determined and from them a transformation rule  $\Phi$  for converting instrument-dependent coordinates ( $x, y, z$ ) into the coordinates ( $X, Y, Z$ ) of the DICOM coordinate system is obtained. Then, for non-instrument-dependent coordinate determination, the instrument coordinates ( $x_p, y_p, z_p$ ) of an imaged point  $P$  are converted by means of the transformation rule  $\Phi$  discovered into non-instrument-dependent coordinates ( $X_p, Y_p, Z_p$ ) of the DICOM coordinate system.

**[0005]** The "Digital Imaging and Communications in Medicine" (DICOM) standard was developed for formatting and exchanging images of medical equipment and integrated in this equipment. DICOM is known, *inter alia*, in the USA, Europe and Japan. On 2nd July 1999 in Virginia, USA, in Supplement 15, the DICOM Committee laid down a standard for images obtained with visible light in endoscopy, microscopy and photography (Supplement 15: Visible Light Image for Endoscopy, Microscopy and Photography). With the present invention, this purely specimen-related and therefore non-instrument-dependent DICOM coordinate system can be implemented on any microscope. The technical solution of the procedure according to the invention comprises two steps. First of all, the microscope coordinate system is calibrated so as to obtain a transformation rule for converting instrument-dependent coordinates into non-instrument-dependent coordinates of the DICOM coordinate system. After this calibration step, the coordinates of any imaged point can be transformed using this transformation rule into non-instrument-dependent coordinates of the DICOM coordinate system. These latter coordinates can then be re-accessed at a later stage or by a different user, even on a different microscope, although obviously the other equipment must contain a calibration facility for the DICOM coordinate system.

**[0006]** For the calibration step, in a particularly advantageous embodiment, a calibration slide is used for setting reference coordinates of the minimum of one reference point  $E_1$ . This calibration slide has calibration crosses on it, marking the set reference points, in accordance with the provisions of the DICOM standard.

**[0007]** In order to be able to give optimum consideration to all transformations in question in the  $(x, y)$  plane, namely translation, rotation and scaling, mathematically at least 2.5 reference points or calibration crosses are needed on the calibration slide. Additional points may be needed if calibration is also to be carried out in direction  $z$ .

**[0008]** As certain types of slides are used in microscopy, it is advantageous to produce a corresponding calibration slide for each type of slide and to use it for the process according to the invention.

**[0009]** For calibration purposes, three calibration crosses corresponding to reference points  $E_1$ ,  $E_2$  and  $E_3$ , for example, are provided on a calibration slide. The  $(X, Y, Z)$  coordinates of these reference or calibration points  $E_1$  to  $E_3$  are fixed. They relate to the zero point of the DICOM coordinate system, which may be located at an outer corner of the slide.

**[0010]** The calibration points  $E_1$  to  $E_n$  ( $n \geq 1$ ) are selected using the microscope stage and the respective  $(x_1, y_1, z_1), \dots, (x_n, y_n, z_n)$  value is received and stored in the native, i.e. instrument-dependent coordinate system of the microscope used. For the calibration points  $E_1$  to  $E_n$  the  $(X, Y, Z)$  values in the DICOM coordinate system and after measurement the  $(x, y, z)$  values in the native coordinate system are known, so that a transformation rule can be calculated using standard methods for converting instrument-dependent coordinates into the non-instrument-dependent coordinates of the DICOM coordinate system.

**[0011]** The transformation method which comes to mind for the  $(x, y)$  coordinates is that of the overdetermined affine transformation. For the transformations of

translation, rotation and scaling by a scaling factor which occur in one plane, mathematically at least 2.5, in practice therefore at least 3, reference points (calibration crosses) are needed if all the above-mentioned calibration possibilities are to be taken into account.

**[0012]** The Z zero point of the DICOM coordinate system is on the surface of the slide (without a glass cover). As the native Z coordinates are also included in the calibration described above, the z value can also be converted into the DICOM coordinate system. In the Z calibration, essentially two cases can be distinguished.

**[0013]** If during calibration z values of the surface of the calibration slide increase or decrease in one direction of the (X, Y) plane, the indication is that the calibration slide is not lying precisely horizontally but constitutes a skewed plane inclined in the Z direction. In this case, to increase the accuracy, Z calibration should also be carried out with an approach in the form of an inclined plane, as otherwise the accuracy of the (X, Y) calibration will be reduced. In this case, the deviation  $\Delta z$  can be measured along the gradient of the inclined plane by focussing on the surface of the slide and then the Z calibration can be carried out, for which mathematically at least 1.5 points are needed. For a Z calibration of this kind together with an overdetermined affine transformation in the (X, Y) plane, therefore, at least 4 points ( $2.5 + 1.5 = 4$ ) are needed on the calibration slide.

**[0014]** If, on the other hand, it is established that the z coordinates of some selected reference points on the calibration slide differ from one another without having the form of an inclined plane, a simple transformation rule would be an averaging process, whereby the average is taken of the above-mentioned z coordinates of the reference points and this average value is defined as the zero point in the z direction. In other words, the average value calculated for the z coordinates corresponds to the zero point in the DICOM coordinate system.

[0015] It is also conceivable for the two above-mentioned effects to occur in combination.

[0016] A calibration slide having at least one reference point with preset reference coordinates in a DICOM coordinate system is proposed for use in the method according to the invention. As already stated, calibration crosses are provided on this calibration slide, constituting the reference points for the method according to the invention. In the DICOM coordinate system the zero point is located on one of the outer corners of the rectangular calibration slide. It is particularly advantageous if the calibration slide corresponds in shape and size to a known type of slide used in microscopy.

[0017] For interoperability it is essential that the calibration is carried out on the respective systems (microscopes) using the method according to the invention. The use of identical calibration slides is most suitable for this purpose.

[0018] As the system for non-instrument-dependent determination of coordinates of a point to be imaged using a microscope which comprises a unit for determining instrument coordinates ( $x_p, y_p, z_p$ ) of an imaged point P, the invention provides a computer unit which calculates, from instrument coordinates ( $x_1, y_1, z_1$ ) of at least one imaged reference point E<sub>1</sub> and associated predetermined object-related reference coordinates (X<sub>1</sub>, Y<sub>1</sub>, Z<sub>1</sub>) in a DICOM coordinate system, a transformation rule  $\Phi$  for converting instrument-dependent coordinates into coordinates of the DICOM coordinate system. The computer unit for calculating the transformation rule may be integrated in the microscope or may be a part of a peripheral computer.

[0019] Using this system according to the invention, instrument-dependent coordinates can be converted into non-instrument-dependent coordinates of the DICOM coordinate system. For this, the transformation rule  $\Phi$  calculated is applied to the coordinates ( $x_p, y_p, z_p$ ) of an imaged point P and the corresponding coordinates (X<sub>p</sub>, Y<sub>p</sub>, Z<sub>p</sub>) are calculated in the non-instrument-dependent DICOM coordinate system. In order

to automate the method according to the invention for calibrating and then calculating non-instrument-dependent coordinates as efficiently as possible, it is useful to implement this method using a computer program which is started up and executed, in particular, on the above-mentioned computer unit of the system according to the invention. The computer program may be stored on data carriers such as CD-ROMs, EEPROMs or in the form of flash memories, or can be downloaded into the working memory through various computer networks, such as Intranet or Internet.

[0020] When this computer program is run, for example after a calibration slide with a DICOM coordinate system has been placed on the microscope stage, the reference points applied in the form of calibration crosses are measured (automatically) in the instrument-dependent coordinate system and the corresponding coordinates are determined. After preferably three or more such reference points have been measured, the computer program begins to calculate the transformation rule. Then a sample is examined using the microscope and the instrument coordinates of a point of interest are automatically converted into non-instrument-dependent coordinates of the DICOM coordinate system by the computer program, using the transformation rule.

[0021] The computer program can control the entire procedure described, by interaction with the user, or automatically execute particular parts of the method in the form of program modules.

[0022] The invention and its advantages will now be described in more detail with reference to an exemplifying embodiment shown in the drawings.

[0023] In the drawings:

[0024] Figure 1 shows a system according to the invention for the non-instrument-dependent determination of coordinates of a point to be imaged with a microscope, in schematic representation;

**[0025]** Figure 2 shows a calibration slide according to the invention and a microscopic image with a schematically shown part of the unit for determining instrument coordinates for calculating the transformation rule  $\Phi$ ;

**[0026]** Figure 3 shows an imaged point P in the instrument-dependent coordinate system and in the DICOM coordinate system.

**[0027]** Figure 1 shows, in highly schematic form, a microscope 1 with objective 7 for the enlarged imaging of structures carried on a slide 6. These structures may be cells or collections of cells but may also be industrial structures such as semiconductor structures. Accordingly, the applications of microscopy extend from the medical field (cytology, histology, pathology) into the industrial field (e.g. wafer production or nanotechnology). In these fields it is essential that any conspicuous features or faults in the structures can be marked and retrieved at a later stage or by a different user.

**[0028]** Frequently, a computer unit 2 is connected to the microscope 1 or to a microscope camera 11 in order to allow microscopic images to be further processed and saved. In the interests of simplicity it will be assumed hereinafter that the microscopic images 8 can be viewed on a monitor 10 of the computer unit 2 and that at least part of the unit 4 for determining instrument coordinates (coordinates in the microscopic image) is also present in the computer unit 2.

**[0029]** The slide 6 is often applied to a microscope stage 5 by vacuum suction, the microscope stage 5 generally being adjustable in its three-dimensional position.

**[0030]** The object structures can be examined using a one-shot or a scan.

**[0031]** The image data transmitted from the microscope 1 or from the microscope camera 11 to the computer unit 2 are displayed, in this embodiment, on the monitor 10 of the computer unit 2 in an instrument-dependent coordinate system, while for example by clicking on a particular point in the display image using a mouse 12 the corresponding

coordinates of this point can be determined and displayed and saved in the microscope system.

**[0032]** In the present case the computer unit 2 has a computer program which is able to derive a transformation rule  $\Phi$  from coordinates of one or more imaged reference points and the associated known preset reference coordinates, based on a DICOM coordinate system, on the slide 6, by means of which equipment-dependent coordinates can be converted into coordinates of a DICOM coordinate system (an example of a computer program of this kind can be found at the end of this specification). It is expedient to calibrate the system by using as the slide a calibration slide 3 with at least one reference point in a DICOM coordinate system, in order to calculate the transformation rule using the reference point or points imaged.

**[0033]** The non-equipment-dependent determination of coordinates of an imaged point which constitutes, for example, a fault, a conspicuous feature or an error, is enormously advantageous for reliably retrieving the point. It enables it to be reliably found in spite of tolerances in the same or similar instruments, e.g. during subsequent monitoring using the same instrument or an instrument of the same construction, but also during later examination on different equipment or in remote microscopy (telepathology or remote diagnosis or operations).

**[0034]** Figure 2 shows a calibration slide 3 with DICOM (XY) coordinate system and the associated microscopic image 8 with instrument-dependent (xy) coordinate system. The transformation rule  $\Phi$  provides the correlation between the two coordinate systems.

**[0035]** The calibration slide 3 according to the invention has six calibration crosses 9 in this embodiment, corresponding to points  $E_1$  to  $E_6$  in a DICOM coordinate system X, Y, Z, while the zero point is in the top left-hand corner of the calibration slide 3. In the interests of simplicity the Z coordinates will not be considered in the following description. Possible options for the Z calibration are mentioned above in the present

description. In order to calibrate the system shown in Figure 1, first of all the calibration slide 3 is placed on the microscope stage 5 and a microscopic image 8 is produced by means of the microscope 1 and the computer unit 2. The bottom half of Figure 2 shows a microscopic image 8 of this kind with an instrument-dependent coordinate system  $x, y$ , while the corresponding instrument coordinates  $(x_1, y_1)$  to  $(x_6, y_6)$  at the calibration crosses shown (E1 to E6) can be determined using a unit 4. It should be mentioned that not all six calibration crosses 9 have to be used for the calibration, but fewer calibration crosses may be sufficient depending on the degree of precision required. As described above, however, it is expedient to use three calibration crosses to derive a transformation rule by overdetermined affine transformation.

[0036] The method of overdetermined affine transformation (cf. the example at the end of this description) is known *per se* and will therefore not be described in detail hereinafter. There are also other possible methods known to the skilled man for deriving the transformation rule  $\Phi$ . The unit 4 for determining instrument coordinates determines the coordinates of a suitable number of imaged calibration crosses, i.e. the corresponding reference points E<sub>1</sub>, E<sub>2</sub>, E<sub>3</sub>..., in the  $x, y$  coordinate system. The coordinates of the corresponding calibration crosses 9 (reference points) on the calibration slide 3 in the DICOM XY coordinate system are prescribed. From these, the computer unit 2 or, more accurately, a corresponding computer program running on this computer unit 2, can calculate the transformation rule  $\Phi$  for converting instrument-dependent coordinates  $(x, y)$  into the coordinates  $(X, Y)$  of the DICOM coordinate system.

[0037] It is useful if calibration slides which correspond to the current slide formats are produced, by means of which associated transformation rules  $\Phi$  can be calculated, as described above.

[0038] Using the transformation rule  $\Phi$  discovered, the instrument coordinates  $(x_p, y_p)$  of an imaged point P, as shown in Figure 3, can now be converted into non-instrument-dependent coordinates  $(X_p, Y_p)$  of the DICOM coordinate system. The point

P may for example represent a conspicuous feature in a cell structure or a fault in a semiconductor structure. The coordinates of the point P are determined by means of the unit 4 for determining instrument coordinates and are converted using the known transformation rule  $\Phi$  into non-instrument-dependent coordinates of the DICOM coordinate system. For subsequent monitoring or re-examination, the non-instrument-dependent coordinates of the point P together with the specimen are submitted for re-examination. The system in which the re-examination is carried out must, of course, also have a calibration facility for the DICOM coordinate system. In particular, this system has to calculate the associated instrument-dependent coordinates of the point P from the DICOM coordinates supplied for the point P using the inverse transformation rule  $\Phi^{-1}$  so that this point can be accessed again in the microscopic image 8.

**[0039]** The following is an example of a computer program written in C programming language, by means of which forward and backward transformation of the coordinates from a native microscope coordinate system and the DICOM coordinate can be carried out using the overdetermined affine transformation method:

```
/*-----*/
// calculation for overdetermined affine transformation
// forward and backward calculation
// coordinate systems are:
//           native microscope-coordinate system, slide-dependent
//           microscope-independent DICOM coordinate system

/*-----*/
#include <stdio.h>

/** Function PROTOTYPES **/
```

```
//given: native microscope coordinates, calculate DICOM coordinates void
CalculateDICOMCoordinates (
```

```
double x_Microscope,
```

```
double y_microscope,
```

```

double *pX_DICOM,
double *pY_DICOM
);

// given: DICOM coordinates, calculate native microscope coordinates void
CalculateNativeMicroscopeCoordinates (
    double X_DICOM,
    double Y_DICOM,
    double *px_Microscope,
    double *py_Microscope
);

// calculate coordinate transformation coefficients for forward and back transformation

int CalcForwBackwTransCoefficients (
    int NoOfGaugingPoints,
    double *x_MicroscopeSystem,
    double *y_MicroscopeSystem,
    double *x_DICOMSystem,
    double *y_DICOMSystem
);

// reset transformation coefficients to default values

void ResetTransformationCoefficients (void);

int CalculateTransformation (
    double *a, double *b, double *c,
    double *d, double *e, double *f,
    int NoOfGaugingPoints,
    double *x_Microscope, double *y_Microscope,
    double *x_DICOM, double *y_DICOM
);

/* static variables for coordinate transformation */

/* forward transformation coefficients */
double aFwd = 1.0;
double bFwd = 1.0;

```

```

double cFwd = 0.0;
double dFwd = 1.0;
double eFwd = 1.0;
double fFwd = 0.0;

/* backward transformation coefficients */
double aBwd = 1.0;
double bBwd = 1.0;
double cBwd = 0.0;
double dBwd = 1.0;
double eBwd = 1.0;
double fBwd = 0.0;

int main (void)
{
    // coordinates of calibration points E1 to E6 in DICOM and native microscope
    system
    double aX_DICOM [] = {3000., 3000., 3000., 17000., 17000., 17000.};
    double aY_DICOM [] = {10000., 30000., 50000, 10000., 30000., 50000.};
    double ax_Microscope [] = {41000., 43000.,45000., 181000., 183000., 185000.};
    double ay_Microscope [] = {129400., 309400., 489400., 126600., 306600.,
    486600.};

    double xMicTest, yMicTest;

    double XDICTest, YDICTest;

    int i;

    printf ("\n\n");

    printf ("Affin Transformation From Native Microscope To DICOM Coordinate
    System\n");
    printf
    ("=====\\n");

    printf ("\n\n\n");

    printf ("Coordinates of gauging points\\n\\n");

    for (i=0; i<=5; ++i) {
        printf ("E%d: X-DICOM=%10.1f Y-DICOM=%10.1f x-Mic=%10.1f y-
        Mic=%10.1f\\n",
        i+1, aXDICOM [i], aY_DICOM [i], axvMicroscope [i], ay_Microscope [i]);

        printf ("\n\n");

        if (0 == CalcForwBackwTransCoefficients (6,

```

```

    ax_Microscope, ay_Microscope, aX_DICOM, aY_DICOM)){ printf
    ("computation failed\n");
    return 0;
}

printf ("forward calculation coefficients\n");
printf ("-----\n");
printf ("X-DICOM = %.4f * x-Mic + %.4f * y-Mic + %.4f\n", aFwd.bFwd.cFwd); printf
("Y-DICOM = %.4f * x-Mic + %.4f * y-Mic + %.4f\n\n\n", dFwd,eFwd,fFwd);

printf ("backward calculation coefficients\n");
printf ("-----\n\n");

printf ("x-Mic = %.4f * X-DICOM + %.4f * Y-DICOM + %.4f\n", aBwd.bBwd.cBwd);
printf ("y-Mic = %.4f * X-DICOM + %.4f * Y-DICOM + %.4f\n\n\n",
dBwd.eBwd.fBwd);

printf ("Tests of calculation\n");
printf (".....\n\n");

printf ("Test1 using microscope coordinates of gauging point E4 as input\n");
printf ("\n\n");

xMicTest = 181000.;
yMicTest = 126600.;

printf ("****input***: x Microscope =%10.1f y Microscope=%10.1f\n", xMicTest,
yMicTest);
CalculateDICOMCoordinates (181000., 126600., &XDICTest, &YDICTest);
printf ("**** result ***: x DICOM =%10.1f y DICOM =%10.1f\n\n", XDICTest
YDICTest);
printf ("End of test1 \n\n");

printf ("Test2 applying forward and backward transformation to test point\n");

printf ("\n\n");

xMicTest =100000.;

yMicTest = 250000.;

printf ("**** input ***: x Microscope =%10.1f y Microscope=%10.1f\n",
xMicTest, yMicTest);
printf ("forward transform\n");

CalculateDICOMCoordinates (xMicTest, yMicTest, &XDICTest, &YDICTest);
printf ("**** result ***: x DICOM =%10.1f y DICOM =%10.1f\n", XDICTest, YDICTest);
printf ("backward transform\n");

xMicTest = 0.;

yMicTest = 0.;

CalculateNativeMicroscopeCoordinates(XDICTest,YDICTest,

```

```

&xMicTest,
&yMicTest);
printf ("*** result **: x Microscope =%10.1f y Microscope =%10.1f\n", xMicTest,
yMicTest);
printf ("End of test2 \n\n");

return 0;

}

/*
// forward transformation: calculate DICOM coordinates from
// native microscope coordinates
*/
void CalculateDICOMCoordinates (
// input: x,y coordinates in the microscope system
double x_Microscope,
double y_Microscope,
//output: X.Y coordinates in the DICOM System
double *pX_DICOM,
double *pY_DICOM
)
{
    *pX_DICOM = aFwd * x_Microscope + bFwd * y_Microscope + cFwd;
    *pY_DICOM = dFwd * x_Microscope + eFwd * y_Microscope + fFwd;

/*
// back transformation: calculate native microscope-coordinates from DICOM
// coordinates
*/
void CalculateNativeMicroscopeCoordinates (
// input: X.Y coordinates in the DICOM System
double X_DICOM,
double Y_DICOM,
//output: x,y coordinates in the microscope system
double *px_Microscope,
double *py_Microscope
)
{
    *px_Microscope = aBwd * X_DICOM + bBwd * Y_DICOM + cBwd;
    *py_Microscope = dBwd * X_DICOM + eBwd * Y_DICOM + fBwd;
}

```

```

/*=====
int CalcForwBackwTransCoefficients (
int NoOfGaugingPoints,

double *x_MicroscopeSystem,
double *y_MicroscopeSystem,
double 'x_DICOMSystem,
double *y_DICOMSystem
=====

if (NoOfGaugingPoints<3) return 0;

if (0 == CalculateTransformation (&aBwd,&bBwd,&cBwd,&dBwd,&eBwd,&fBwd,
NoOfGaugingPoints, x_MicroscopeSystem, y_MicroscopeSystem,
x_DICOMSystem, y_DICOMSystem))

return 0;

if (0 == CalculateTransformation (&aFwd,&bFwd,&cFwd,&dFwd,&eFwd,&fFwd,
NoOfGaugingPoints, x_DICOMSystem, y_DICOMSystem,
x_MicroscopeSystem, y_MicroscopeSystem))

return 0;

return 1;
}

=====

void ResetTransformationCoefficients (void)
=====

/* reset coefficients */
aFwd = 1.0;
bFwd = 1.0;
cFwd = 0.0;
dFwd = 1.0;
eFwd = 1.0;
fFwd = 0.0;
aBwd = 1.0;
bBwd = 1.0;
cBwd = 0.0;
dBwd = 1.0;
eBwd = 1.0;
fBwd = 0.0;
}

=====

Int CalculateTransformation ( double *a, double *b, double *c,
double *d, double *e, double *f,
int NoOfGaugingPoints,
double *x_Microscope, double *y_Microscope,
double *x_DICOM, double *y_DICOM

```

```

        )
/*=====*/
{
    int i;

    double *xDIC, *yDIC, *xMic, *yMic;
    double r1, r2, r3, r4, r5, r6, r7, r8, double r9, r10,
    r11, r12, r13, r14, r15;

    r1 = r2 = r3 = r4 = r5 = r6 = r7 = r8 = r13 = r14 = r15 = 0.0;

    xDIC = x_DICOM; yDIC =
    y_DICOM; xMic =
    x_Microscope; yMic =
    y_Microscope;

    for (i = 0; i < NoOfGaugingPoints; i++)
    {
        r1 += *xDIC;
        r2 += *yDIC;
        r3 += *xDIC * *xDIC; r4 +=
        *yDIC * *yDIC; r5 +=
        *xMic;
        r6 += *xDIC * *yDIC; r7 +=
        *xMic * *yDIC; r8 += *xDIC
        * *xMic++;
        r13 += *yMic;
        r14 += *yDIC++ * *yMic;
        r15 += *xDIC++ * *yMic++;
    }

    /* accounting of coefficients a, b, c */

    for (i=1; i<=2; i++) {
        r9 = r3 * r4 * NoOfGaugingPoints + 2 * r1 * r2 * r6 - r1 * r1 * r4 - r2 * r2 * r3 \
        - r6 * r6 'NoOfGaugingPoints;
        /* transformation is singular */
        if (r9 == 0.0)
            return 0;

        if (i == 2)
        {
            /* accounting of coefficients d, e, f */
            r5 = r13;
            r7 = r14;
            r8 = r15;
        }
    }
}

```

```

        }
        r10 = r8 *r4 * NoOfGaugingPoints + r6 * r2 *r5 +r1 * r7 * r2 -\
                           r1 * r4 * r5 - r8 * r2 * r2 - r6 * r7 *
NoOfGaugingPoints;
        r11 = r3 *r7 * NoOfGaugingPoints + r8 * r2 *r1 +r1 * r6 * r5 - r1 * r1 * r7 - r3 * r2 \
                           * r5 - r8 * r6 *
NoOfGaugingPoints ; r12 = r3 *r4 * r5 + r6 * r7 *r1 +r8 * r6 * r2 \
                           - r8 * r4 * r1 - r5 * r6 * r6 - r3 * r7 * r2;
        r10 = r10/r9;
        r11 =r11 /r9;
        r12/=r9;
        if(i==1)
        {
            *a = r10;  *b = r11; *c = r12;
        }
        else
        {
            *d = r10;  *e = r11; *f = r12;
        }
    }
    return 1;
}

/*
===== End (end of file)=====

```

**[0040]** The following text shows a printout as generated by the above program when six calibration points E1 to E6 are preset and then two tests are carried out. The first test (Test1) carries out a back-transformation into the DICOM coordinate system for calibration point E4, whereas the second test (Test2) carries out forward and back transformation of a given test point (P):

#### Affine Transformation From Native Microscope To DICOM Coordinate System

---



---

#### Coordinates of gauging points

E1: X-DICOM= 3000.0 Y-DICOM= 10000.0 x-Mic= 41000.0 y-Mic= 129400.0

E2: X-DICOM= 3000.0	Y-DICOM= 30000.0	x-Mic= 43000.0	y-Mic= 309400.0
E3: X-DICOM^ 3000.0	Y-DICOM= 50000.0	x-Mic= 45000.0	y-Mic= 489400.0
E4: X-DICOM= 17000.0	Y-DICOM= 10000.0	x-Mic= 181000.0	y-Mic= 126600.0
E5: X-DICOM= 17000.0	Y-DICOM= 30000.0	x-Mic= 183000.0	y-Mic= 306600.0
E6: X-DICOM= 17000.0	Y-DICOM= 50000.0	x-Mic= 185000.0	y-Mic= 486600.0

forward calculation coefficients

---

$$X-DICOM = 0.1000 * x-Mic + -0.0011 * y-Mic + -955.3433$$

$$Y-DICOM = 0.0022 * x-Mic + 0.1111 * y-Mic + -4465.6743$$

backward calculation coefficients

---

$$x-Mic = 10.0000 * X-DICOM + 0.1000 * Y-DICOM + 10000.0000$$

$$y-Mic = -0.2000 * X-DICOM + 9.0000 * Y-DICOM + 40000.0000$$

Tests of calculation

---

Test1 using microscope coordinates of gauging point E4 as input

\*\*\* input \*\*\*: x Microscope = 181000.0 y Microscope= 126600.0 \*\*\*

result \*\*\*: x DICOM = 17000.0 y DICOM = 10000.0

End of test1

Test2 applying forward and backward transformation to test point

\*\*\* input \*\*\*: x Microscope = 100000.0 y Microscope= 250000.0

forward transform

\*\*\* result \*\*\*: x DICOM = 8764.7 y DICOM = 23528.1

backward transform

\*\*\* result \*\*\*: x Microscope = 100000.0 y Microscope = 250000.0

End of test2

[0041] The printout of the above text produced by the program is reproduced below in German (as far as possible) for ease of understanding:

Affine Transformation vom nativen Mikroskop- zum DICOM Koordinatensystem

Koordinaten der Eichpunkte

E1:X-DICOM= 3000.0	Y-DICOM= 10000.0	x-Mic= 41000.0	y-Mic= 129400.0
E2: X-DICOM= 3000.0	Y-DICOM= 30000.0	x-Mic= 43000.0	y-Mic= 309400.0
E3: X-DICOM= 3000.0	Y-DICOM= 50000.0	x-Mic= 45000.0	y-Mic= 489400.0
E4:X-DICOM= 17000.0	Y-DICOM= 10000.0	x-Mic= 181000.0	y-Mic= 126600.0
E5:X-DICOM= 17000.0	Y-DICOM= 30000.0	x-Mic= 183000.0	y-Mic= 306600.0
E6:X-DICOM= 17000.0	Y-DICOM= 50000.0	x-Mic= 185000.0	y-Mic= 486600.0

Berechnungskoeffizienten für Hintransformation

$$X-DICOM = 0.1000 * x-Mic + -0.0011 * y-Mic + -955.3433$$

$$Y-DICOM = 0.0022 * x-Mic + 0.1111 * y-Mic + -4465.6743$$

Berechnungskoeffizienten für Rücktransformation

$$x-Mic = 10.0000 * X-DICOM + 0.1000 * Y-DICOM + 10000.0000 y-$$

$$Mic = -0.2000 * X-DICOM + 9.0000 * Y-DICOM + 40000.0000$$

Berechnungstests

Test1 unter Verwendung der Mikroskopkoordinaten des Eichpunktes E4 als Eingabe

\*\*\* Eingabe \*\*\*: x Microscope = 181000.0 y Microscope= 126600.0

\*\*\* Ergebnis \*\*\*: x DICOM = 17000.0 y DICOM = 10000.0

Ende des Test1

Test2 mit Hin- und Rücktransformation des Testpunktes

\*\*\*Eingabe \*\*\*: x Microscope = 100000.0 y Microscope= 250000.0

Hintransformation

\*\*\* Ergebnis \*\*\*: x DICOM = 8764.7 y DICOM = 23528.1

Rücktransformation

\*\*\* Ergebnis \*\*\*: x Microscope = 100000.0 y Microscope = 250000.0

Ende des Test2

## List of reference numerals

[0042]	1	Microscope
[0043]	2	Computer unit
[0044]	3	Calibration slide
[0045]	4	Unit for determining instrument coordinates
[0046]	5	Microscope stage
[0047]	6	Slide
[0048]	7	Objective
[0049]	8	Microscopic image
[0050]	9	Calibration crosses, points on slide
[0051]	10	Monitor
[0052]	11	Camera
[0053]	12	(Computer) mouse
[0054]	P	Imaged point
[0055]	$\Phi$	Transformation rule
[0056] coordinates	X, Y, Z	Coordinates in the DICOM coordinate system, reference
[0057]	x, y, z	Coordinates in the microscope system, instrument coordinates
[0058]	E <sub>1</sub> , ..., E <sub>6</sub>	Calibration crosses, points, reference points